



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

APLIKACE PRO SLEDOVÁNÍ ALKOHOLU V KRVÍ

AN APPLICATION FOR TRACKING OF ALCOHOL IN BLOOD

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

ARTHUR NÁCAR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN HRUBÝ, Ph.D.

BRNO 2021

Zadání bakalářské práce



Student: **Nácar Arthur**

Program: Informační technologie

Název: **Aplikace pro sledování alkoholu v krvi**
An Application for Tracking of Alcohol in Blood

Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte programování aplikací pro iOS a pro watchOS. Prostudujte knihovnu HealthKit. Prostudujte metodiky posuzování odbourávání alkoholu z krve v závislosti na zdravotním stavu a pohybových aktivitách.
2. Navrhněte aplikaci pro sběr dat o konzumovaném alkoholu uživatele. Navrhněte algoritmy predikce budoucího poklesu koncentrace alkoholu v krvi podle zaznamenávaných dat. Zaměřte se na vizualizaci predikcí.
3. Aplikaci implementujte pro Apple Watch a iPhone.
4. Aplikaci testujte se zapojením více uživatelů a predikce srovnávejte s objektivním měřením alkoholtesterem.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Hrubý Martin, Ing., Ph.D.**

Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.

Datum zadání: 1. listopadu 2020

Datum odevzdání: 12. května 2021

Datum schválení: 11. listopadu 2020

Abstrakt

Cílem této práce je vytvořit mobilní aplikaci pro zařízení iOS a aplikaci pro chytré hodinky se systémem WatchOS, která uživateli umožní sledování koncentrace alkoholu v krvi v průběhu času. Implementace proběhla v jazyce Swift s využitím frameworku Apple HealthKit, pro získávání osobních dat uživatele. Predikce koncentrace etanolu v krvi jsou získávány za použití simulačního modelu, který zohledňuje resorpční fázi etanolu a jako základ fáze eliminace používá rozšířenou Widmarkovu metodu. Výsledkem této práce je aplikace, která uživateli poskytne informaci o době potřebné k vystřízlivění s detailním přehledem průběhu koncentrace alkoholu v krvi za použití vícero dostupných metrik.

Abstract

The aim of this thesis was to create mobile applications for iOS devices and a smartwatches with WatchOS that allows the user to monitor blood alcohol content over time. The application is implemented in Swift language and it uses Apple HealthKit framework to access the user's personal data. Predictions of blood ethanol content are calculated using a simulation model that takes into account the resorption phase of ethanol and uses the extended Widmark method to estimate the course of the elimination phase. The result of this work is an application that provides the user with information about the time needed to get a sober and detailed overview of the course of blood alcohol concentration using several available metrics.

Klíčová slova

iOS, Apple, iPhone, mobilní aplikace, HealthKit, predikce alkoholu, WatchOS, WatchKit, Swift, Widmarkova metoda.

Keywords

iOS, Apple, iPhone, mobile application, GPS, HealthKit, alcohol dilution, WatchOS, WatchKit, Swift, Widmark equation.

Citace

NÁCAR, Arthur. *Aplikace pro sledování alkoholu v krvi*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Hrubý, Ph.D.

Aplikace pro sledování alkoholu v krvi

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Martina Hrubého, Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....

Arthur Nácar
11. května 2021

Poděkování

Rád bych poděkoval svému vedoucímu práce panu Ing. Martinovi Hrubému, Ph.D. za čas, který obětoval vedení mé práce, poskytnuté odborné rady a trpělivost. Dále bych také rád poděkoval mým rodičům za velkou podporu při studiu.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 3 |
| 2 | Kontext vývoje aplikace | 4 |
| 2.1 | Modelování koncentrace alkoholu v krvi | 4 |
| 2.1.1 | Fáze vstřebávání | 4 |
| 2.1.2 | Fáze odbourávání | 4 |
| 2.1.3 | R faktor | 5 |
| 2.1.4 | Rychlost eliminace | 5 |
| 2.2 | Analýza existujících aplikací | 6 |
| 2.3 | Vývoj aplikací pro iOS a WatchOS | 7 |
| 2.3.1 | Vývojové prostředí XCode | 7 |
| 2.3.2 | Apple Human Interface Guidelines | 8 |
| 2.3.3 | Apple Health a HealthKit | 8 |
| 2.3.4 | WatchKit | 9 |
| 2.3.5 | Watch Connectivity | 10 |
| 3 | Návrh a implementace | 11 |
| 3.1 | Návrh aplikace | 11 |
| 3.1.1 | Návrh aplikace iOS | 11 |
| 3.1.2 | Návrh aplikace WatchOS | 12 |
| 3.1.3 | Návrh barevné palety | 12 |
| 3.1.4 | Prototypování | 13 |
| 3.2 | Uživatelské rozhraní iOS | 14 |
| 3.2.1 | Hlavní obrazovka | 14 |
| 3.2.2 | Obrazovka vkládání nápojů | 15 |
| 3.2.3 | Obrazovka volby objemu nápoje | 15 |
| 3.2.4 | Obrazovka Kalendář | 16 |
| 3.2.5 | Obrazovka Náhled dne | 16 |
| 3.2.6 | Obrazovka osobního profilu | 17 |
| 3.3 | Implementace ovládacích prvků | 17 |
| 3.3.1 | Analogové hodiny | 17 |
| 3.4 | Uživatelské rozhraní WatchOS | 19 |
| 3.4.1 | MainInterface | 19 |
| 3.4.2 | FavouriteInterface | 19 |
| 3.4.3 | AllDrinksInterface | 20 |
| 3.5 | Datový model | 20 |
| 3.5.1 | Seznam nápojů | 21 |
| 3.5.2 | Záznamy o konzumaci | 22 |

| | | |
|----------|--|-----------|
| 3.5.3 | Oblíbené nápoje | 22 |
| 3.6 | Implementace simulačního modelu | 22 |
| 4 | Testování | 24 |
| 4.1 | Testování uživatelského rozhraní | 24 |
| 4.2 | Ověření platnosti na základě reálného měření | 25 |
| 4.2.1 | Pokus č. 1 | 25 |
| 4.2.2 | Metodika | 25 |
| 5 | Závěr | 29 |
| | Literatura | 30 |
| A | Obsah přiloženého CD | 32 |

Kapitola 1

Úvod

Je obecně nastoleným trendem, že se lidé snaží získat přehled o svém těle a o svém zdraví za pomoci moderních technologií. Necháváme si sledovat svou fyzickou aktivitu, měříme naše životní funkce a řídíme se doporučeními, která pro nás mobilní zařízení vytváří. Mobilní telefon má o nás lepší přehled než náš lékař – také díky tomu, že jej bereme takřka všude s sebou. Aspekt našeho zdraví, na který není kladen dostatečný zřetel, je spojen s konzumací alkoholu.

Konzumace alkoholu je dnes běžným jevem ve společnosti. Česká republika se dlouhodobě drží na předních pozicích ve spotřebě alkoholu na člověka za rok. Téměř každá osoba starší 18 let má zkušenosti s jeho konzumací a stále se snižuje věková hranice prvouživatelů. Přes tyto bohaté zkušenosti, které jako společnost s konzumací máme, nejsme dobře schopni předcházet neblahým vlivům na naše zdraví, mezilidské vztahy a osudy. Přeceňování svých sil, bagatelizování problémů spojených s alkoholem či dokonce jejich odmítání, může mít mnohdy fatální následky. Já bych svou práci rád přispěl přinejmenším k jejich zmírnění. Pokud by problémům pomáhala dokonce předcházet, byl bych nadmíru spokojen.

Tato práce popisuje vývoj mobilní aplikace AlkoKolik pro systém iOS a jejího derivátu pro chytré hodinky se systémem WatchOS, která má sloužit jako nástroj pro sběr informací o konzumaci alkoholických nápojů a ve spolupráci s daty ze zdravotní databáze Apple Health aplikace vytvářet predikci koncentrace alkoholu v krvi. Aplikace následně uživateli prezentuje odpovědi na otázky „kdy dojde k vystřízlivění?“, „jak se pohybovala koncentrace alkoholu v krvi během dne?“ a „zdali zjištěné chování je rizikové vzhledem ke vzniku závislosti na alkoholu?“.

V kapitole 2, v rámci uvedení do problematiky, vysvětlím, jak probíhá metabolismus alkoholu v těle a jaké jsou ovlivňující faktory, kterými se řídí tento proces. Následně se zaměřím na současnou nabídku podobných aplikací, zabývajících se podobnou problematikou, a osvětlím technologie, které jsou využívány k vývoji mobilních aplikací pro platformy iOS a WatchOS. Kapitola 3 se bude zabírat návrhem aplikace a implementací navrhovaného řešení. V poslední kapitole 4 osvětlím, jak probíhalo testování aplikace a jejího predikčního modelu a jaké jsou uživatelské zkušenosti. Nakonec dojde ke zhodnocení naplnění cílů, které jsem si stanovil při vývoji této aplikace.

Kapitola 2

Kontext vývoje aplikace

2.1 Modelování koncentrace alkoholu v krvi

Obecně popisem distribuce návykových látek, ale i léčiv, se zabývá vědní obor s názvem farmakokinetika [6]. Většina drog a alkohol z toho nevyjímaje se chovají podobným způsobem. V praxi se většina aplikací modelující obsah alkoholu v krvi zabývá pouze jeho aktuální hodnotou, která koluje v žilách, případně modelují jeho eliminaci z krve na základě jeho aktuální hladiny.

2.1.1 Fáze vstřebávání

Ke vstřebávání alkoholu z nápoje dochází bezprostředně po jeho požití. Mimo dutinu ústní tak dochází ke vstřebávání v žaludku, a především ve střevě, kde je vstřebáno 65% – 70% celé dávky. Proces lze popsat kinetikou chemické reakce prvního řádu. To znamená, že rychlost absorpce látky je přímo úměrná její koncentraci v těle.

Na základě této znalosti lze vztah vyjádřit množstvím absorbované látky o hmotnosti m v čase t matematickou rovnicí ve tvaru (2.1) [9], kde k je rychlostní konstanta.

$$m_{absor.} = m_{poit} - (1 - e^{-tk}) \quad (2.1)$$

Tato konstanta se může velmi lišit. Záleží například na tom, jestli jedinec pije po jídle, nebo takzvaně na lačno, či jaký druh nápoje pije. Obecně sekty, které jsou sycené oxidem uhličitým, se vstřebávají do krve mnohem rychleji. To je dáno tím, že oxid uhličitý obsažený v nápoji dráždí stěnu žaludku, která se více prokrví a umožní rychlejší prostup alkoholu z trávicího traktu do krevního řečiště. Obecně lze ale říct, že dávka alkoholu se vstřebá v časovém rozmezí třiceti minut až dvou hodin. [9]

2.1.2 Fáze odbourávání

K eliminaci alkoholu z krve dochází konstantní rychlostí, nebo také se dá říct, že eliminace se řídí pravidly kinetiky nultého řádu. Tento objev učinil Widmark (1932) a na základě svého výzkumu vytvořil první matematický model předpovídající průběh alkoholemické křivky [15]. Dnes je Widmarkův vzorec (2.2) všeobecně používaný v soudně-lékařské praxi pro výpočet množství alkoholu v krvi a také tvoří základ modelu průběhu alkoholu v krvi mé aplikace [6].

$$C = \frac{m}{rW} - \beta t \quad (2.2)$$

Slovně by se dalo interpretovat, že koncentrace alkoholu v krvi C (v gramech na kilogram neboli promile) v čase t je závislá od množství m (gramů) čistého alkoholu, r faktoru, eliminačního faktoru β (jejichž význam popíšu v následujících dvou podkapitolách 2.1.3, 2.1.4) a tělesné hmotnosti W (v kg)

2.1.3 R faktor

Widmark při svém výzkumu koncentrace alkoholu v krvi zjistil, že naměřená koncentrace alkoholu je vždy vyšší, než by se dalo předpokládat podle klasického vztahu koncentrace rozpuštěné látky v celku, která je dána rovnicí m/W , kdy m je množství alkoholu v gramech, které se rozpouští v těle o hmotnosti W udávané v kilogramech.

Tento vztah by platil pouze v případě, že by všechny tkáně absorbovaly alkohol stejně. Jelikož je alkohol dokonale mísitelný v tekutinách, tak se schopnost absorpce tkáně odvíjí od podílu vody obsažené. Kostí a tukové buňky obsahují velmi malé procento vody, a tudíž dokážou absorbovat pouze stopové množství alkoholu, zatímco krev, ve které byla naměřená velmi vysoká koncentrace alkoholu, obsahuje převážnou většinu vody.

Aby tento nepoměr mohl být zohledněn ve vzorci pro výpočet výsledné koncentrace, byl navrhnut takzvaný r faktor, nebo též Widmarkův faktor. Jeho jednotkou jsou litry na kilogram a vyjadřuje podíl vody v krvi ku celkové koncentraci vody v jedinci. Velikost podílu je především závislá od pohlaví, věku a indexu BMI.

Ženy mají tento faktor zpravidla mnohem nižší než muži. Tato rozdílnost je dána jejich odlišnou konstitucí těla, kdy průměrný vzrůst u žen je mnohem menší než u mužů, a zároveň ženské tělo obsahuje větší procento tukové tkáně [3].

Pro výpočet r faktoru existuje vícero metodik. Ve své aplikaci jsem pro výpočet zvolil pět nejvýznamnějších způsobů, které vznikly na základě statistických měření na různých početných vzorcích populace [7].

Tabulka 2.1: Tabulka metod pro výpočet faktoru r

| Metoda | Muži | Ženy |
|------------|---|--|
| Watson[14] | $0.39834 + \frac{12.725 \times h}{w} - \frac{0.11275g}{h^2} + \frac{2.8993}{w}$ | $0.29218 - \frac{12.666 \times h}{w} - \frac{2.4846}{w}$ |
| Forrest[4] | $1.0178 - \frac{0.12127 \times w}{h^2}$ | $0.8736 - \frac{0.0124 \times w}{h^2}$ |
| Seidel[11] | $0.31608 - 0.004821 \times w + 0.4632 \times h$ | $0.31223 - 0.006446 \times w + 0.4466 \times h$ |
| Ulrich[13] | $0.715 - 0.00462 \times w + 0.22 \times h$ | |

Kde:

w : je hmotnost v kilogramech

h : je výška v

g : je věk

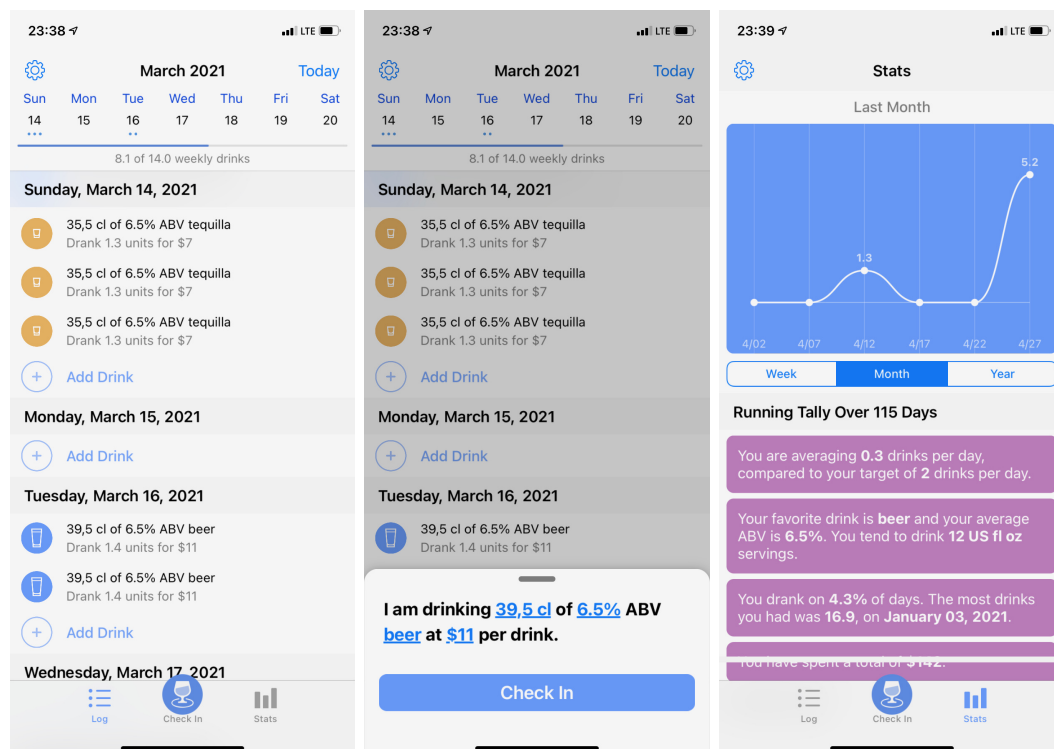
2.1.4 Rychlost eliminace

Rychlost eliminace alkoholu z krve mívá označení β nebo též *beta60* a její jednotkou je gram za hodinu. Její přesnou hodnotu nelze jednoznačně určit, jelikož se může u každého jedince výrazně lišit. Je závislá na faktorech, které nejsme schopni jednoznačně určit, jako je

hmotnost a zdravotní stav jater. Ze statistických měření, které provedl Widmark a která jsou všeobecně přijímána odbornou veřejností, však vyplývá, že rychlost eliminace se pohybuje mezi 0.009 až 0.035 gramů za hodinu s průměrnou hodnotou 0.018 gramů. Při měření aktuální koncentrace alkoholu v krvi se v praxi používají hodnoty v rozmezí od 0.017 do 0.020 gramů za hodinu [6].

2.2 Analýza existujících aplikací

Good Spirits



Obrázek 2.1: Uživatelské rozhraní aplikace *Good Spirits*

Aplikace, která je podobná svou tematikou konzumace alkoholu, a zároveň se nachází v App Store¹ nese název *Good Spirits*². Svým zaměřením je spíše určena ke sledování dlouhodobých návyků. Uživatel v průběhu času zaznamenává druh, objem, obsah alkoholu a cenu nápoje, který zkonsumoval. Aplikace následně vyhodnocuje detailní statistiky o jeho návycích a také útratě. Aplikace obsahuje integraci Apple HealthKit (2.3.3) a dokáže do zdravotní databáze zapsat přijaté kalorie. Neposkytuje však zpětnou vazbu o aktuálním stavu podnapilosti uživatele. Svou propracovaností se jedná spíše o studentský projekt než o komerční produkt.

The image shows the Drunk Calc web application interface, divided into two main steps: Step 1: Personal Details and Step 2: Drink Details. Step 1 includes fields for Biological Sex (Male/Female), Weight (kg) with a 'Change to lbs' link, and Height (cm) with a 'Change to ft & in' link. Step 2 includes fields for Drink Type (Beer, Wine, Cider, Spirits, Cocktail, Custom), Drink Size (mL) (Glass 285mL, Bottle 375mL, Pint 570mL, Custom), Drink Strength (%) (Light 2.7%, Mid 3.5%, Full 4.8%, Custom), and Hunger Level (Very Hungry, Hungry, Not Hungry, Full). To the right, there is a BAC Table showing Current BAC (0.00%), Peak BAC (0.13%), Time Till Peak (36 minutes), and Time Till Zero (9.0 hours). Below the table is a 'Get Home Safely' message. At the bottom right is a BAC Chart showing a line graph of BAC over time, starting at 1:15 PM and ending at 10:15 PM on the 28th of April.

Obrázek 2.2: Uživatelské rozhraní webové aplikace *Drunk Calc*

Drunk Calc

Druhou užitečnou aplikací, kterou bych rád zmínil je *Drunk Calc*³. V tomto případě se jedná o webovou aplikaci, nikoli o aplikaci mobilní. *Drunk Calc* je, narozdíl od ostatních běžných alkohol kalkulaček, vylepšená o schopnost predikovat vývoj alkoholemické křivky, která ve svém modelu zohledňuje resorpční fázi (2.1.1). Uživatel je vyzván, aby vybral jeden ze čtyř stavů, který nejlépe popisuje, jak moc se cítí být sytý. Na základě této informace je upravena délka resorpční fáze. Tělesné míry mohou být zadány jak v metrických, tak i v imperiálních jednotkách. Minimalisticky koncipovaný vzhled popisované aplikace je možné vidět na obrázku 2.2.

2.3 Vývoj aplikací pro iOS a WatchOS

2.3.1 Vývojové prostředí XCode

XCode je vývojové prostředí od společnosti Apple, které usnadňuje vývoj aplikací pro platformy spadající do ekosystému Applu. Jeho distribuce je poskytována bezplatně všem uživatelům operačního systému macOS, kteří jsou registrováni do vývojářského programu.

Součástí prostředí je překladač jazyka Swift a Objective-C. Zkompilovaný projekt lze následně okamžitě spustit ve vestavěném simulátoru virtuálních zařízení iOS/WatchOS.

¹obchod Apple App Store – <https://www.apple.com/cz/app-store/>

²aplikace Good Spirits – <https://apps.apple.com/cz/app/good-spirits/id1434237439>

³webová aplikace Drunk Calc – <https://drunkcalc.com>

Samozřejmou součástí moderního IDE⁴, jako je *XCode*, je vestavěný debugger a zabudovaná podpora verzování. Nespornou výhodou programování aplikací v prostředí *XCode* je fakt, že je skrze vývojové prostředí možné aplikaci nainstalovat, testovat aplikaci, a dokonce i krokovat jednotlivé příkazy přímo na fyzickém zařízení.

Hlavním konkurenčním nástrojem je *AppCode*⁵ od JetBrains. Ten však na rozdíl od *XCode* nenabízí funkci Interface Builderu, což je grafické prostředí pro vytváření UI aplikace. Vývojář v prostředí Interface Builder může skládat jednotlivé prvky uživatelského rozhraní přímo do Views, navzájem je řetězit a tvořit celou vizuální hierarchii aplikace.

2.3.2 Apple Human Interface Guidelines

Jedná se o soubor pravidel a doporučení společnosti Apple, které by měla kvalitní aplikace pro iOS/WatchOS splňovat. Základní sada doporučení pro vývoj aplikace zní[2]:

- Aplikace by měla být ve svém vizuálním stylu a způsobu ovládání konzistentní se zbytkem systému, aby bylo uživateli na první pohled jasné, jak se aplikace ovládá bez nutnosti učení se složitých postupů.
- Vzhled aplikace by měl odrážet účel, ke kterému byla stvořena. V případě, že je aplikace navržena ke splnění náročnějších úkolů vyžadující zvýšené soustředění, je nežádoucí do ní implementovat výrazné ovládací prvky, které by mohly odvádět pozornost od plnění úkolu.
- Uživatel by měl dostávat zpětnou vazbu na provedené akce a zároveň by se změny měly zobrazovat ihned a automaticky po jejich provedení bez nutnosti zásahu uživatele. Například vložení nového prvku do databáze by mělo automaticky invokovat překreslení tabulky. Dalším příkladem by mohlo být stlačení tlačítka, přičemž by se jako zpětná vazba měla zobrazit animace stlačení, aby uživateli bylo jasné, že akce byla zaznamenána.

2.3.3 Apple Health a HealthKit

V roce 2014 byl představen vývojářský framework HealthKit. Účelem frameworku je vytvořit jednotné rozhraní pro čtení, vkládání a správu zdravotních a fitness dat uživatele. Data lze následně souhrnně zobrazit v aplikaci Apple Zdraví na zařízeních iPhone/iPod.[1] Vývojář nemusí implementovat způsob, jakým se budou zaznamenána data sdílet mezi ostatními aplikacemi a zároveň má jistotu, že požadovaná data jsou aktuální a správně převedená na měrnou jednotku, která je adekvátní dané situaci.

Uživatelská data jsou řazena do několika skupin reprezentovaných datovými typy `HKCharacteristicType`, `HKSampleType`. `HKCharacteristicType` reprezentuje informace o uživateli, které jsou zpravidla neměnné, jako je pohlaví, datum narození, či krevní skupina. Ke změně nemůže dojít prostřednictvím aplikace třetí strany. Jejich úpravu může provést uživatel pouze manuálně v aplikaci Apple Zdraví.

Druhým typem uživatelských dat jsou záznamy o měření neboli `HKSampleType`. Tyto data se vztahují k nějakému časovému horizontu, protože tento údaj může být zásadní pro jejich relevanci. Mohou být jak kvantitativní, tak i kvalitativní povahy. Typickým příkladem je váha, kde naměřená hodnota nemusí být za pár hodin aktuální a dává smysl ji opakovaně zaznamenávat v čase.

⁴IDE – Integrated Development Environment

⁵JetBrains AppCode – <https://www.jetbrains.com/objc>

Jednotlivé záznamy daného typu reprezentuje objekt typu `HKSample`, který mimo výše zmíněné údaje o časové platnosti také obsahuje informaci o původci záznamu – zdali údaj vložil uživatel, nebo byl přidán nainstalovanou aplikací.

Hlavním prvkem frameworku je objekt `HKHealthStore`, který je v rámci aplikace singletonem a reprezentuje vstupní bod do zdravotní databáze. Při každém spuštění aplikace by měla být ověřena dostupnost metodou `isHealthDataAvailable`, abychom vyloučili pokus o přístup do databáze ze zařízení, které funkci Apple Health nepodporuje, jako je třeba iPad. Součástí jsou metody `save`, `execute`, `delete`, obstarávající CRUD⁶ operace nad databází.

Jelikož se všechna zdravotní data řadí mezi citlivé údaje, každá aplikace používající `HealthKit` musí z bezpečnostních důvodů explicitně žádat uživatele o povolení přístupu k záznamům daného typu metodou `requestAuthorization`, jež je součástí `HKDataStore`. V případě, že uživatel neschválí přístup k takovýmto datům, framework se poté při dotazování chová, jako by žádné záznamy neexistovaly.

2.3.4 WatchKit

WatchKit je framework tvořící základ pro aplikace na systému WatchOS. Obsahuje komponenty, které obstarávají správu procesů na pozadí, zajišťují přístup k systémovým nástrojům a charakteristickým údajům zařízení a také obsahuje komponenty mající na starost zobrazování uživatelské rozhraní. Každá aplikace pro hodinky se skládá z dvou částí „WatchKit extension“, která obsahuje veškerou aplikační logiku naprogramovanou vývojářem, a „WatchKit App“, která obstarává zobrazení prvků uživatelského rozhraní. Součástí „WatchKit App“ je Storyboard a veškerý statický vizuální obsah, jako jsou obrázky, předdefinované barvy apod., které jsou potřebné pro správné vykreslení obrazovky aplikace.

Obsluhu uživatelského rozhraní obstarávají kontrolery implementované vývojářem, které jsou odvozené od třídy `WKInterfaceController`. Uživateli na obrazovce v jednu chvíli může být prezentovat pouze jeden kontroler podobně, jako tomu je u aplikacích pro iOS. `WKInterfaceController` obsahuje několik metod, které jsou spouštěny v různých etapách aplikačního cyklu.

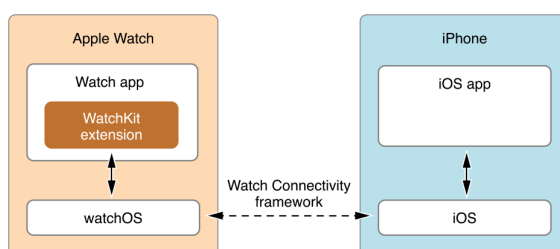
- **awake** – Tato metoda je volána po inicializaci kontroleru a se používá k dokončení inicializace dat, která potřebují pro svou existenci kontext. Například nastavují se zde hodnoty pro `WKInterfaceLabel`, zdrojová data pro `WKInterfaceTable`, určuje se delegátsví.
- **willActivate** – Metoda je volána těsně před tím, než se interface dostane do popředí. Měla by sloužit pro provádění drobných úprav a změn.
- **didDeactivate** – Volá se při odebrání pohledu z aplikační hierarchie. Měla by sloužit k přípravě interface k přechodu do stavu na pozadí
- **didAppear** – Metoda je volána, když se interface stane viditelným na obrazovce. Slouží k zobrazování animací a obsluze dalších úkonů spojených s obsluhou UI.
- **willDisappear** – Voláno v případě, že interface přestává být v popředí. V této metodě by mělo docházet k ukládání perzistentních dat, spouštění animací spojených s ukončením zobrazování interface.

⁶CRUD – Create, Read, Update, Delete

2.3.5 Watch Connectivity

Tento framework zajišťuje komunikaci a přenos dat mezi iOS aplikací a WatchKit extension z ní odvozené na spárovaných Apple Watch. Hlavním předpokladem pro úspěšný přenos informací je vytvoření instance třídy `WCSession` a určení delegátní třídy, která bude zpracovávat příchozí požadavky a volání metody `WCSession.activate()`, která zahájí spojení. Po úvodním ustanovení spojení je možné na spárované zařízení posílat zprávy metodou `sendMessage` a příjemce na ně bude reagovat v těle metody `session`, která je nutnou součástí delegáta. V rámci frameworku jsou zprávy implementovány pomocí slovníku, kde se pod předem dohodnutý klíč ukládají data, která jsou odeslána příjemci.

Jelikož jsou chytré hodinky typem zařízení, kde je kladen velký důraz na úsporu energie, mobilní aplikace nemůže iniciovat komunikaci, pokud aplikace na hodinkách není aktivní v popředí. Naopak však lze spouštět výpočty, i když je aplikace na iPhone spuštěná pouze v pozadí.



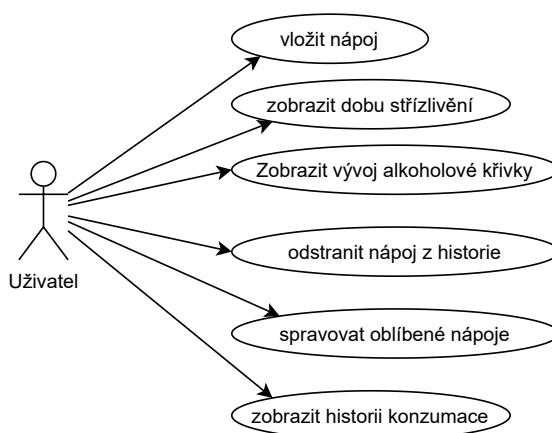
Obrázek 2.3: Diagram komunikace mezi hodinkami a iOS aplikací pomocí knihovny Watch Connectivity, převzato z [5]

Kapitola 3

Návrh a implementace

3.1 Návrh aplikace

3.1.1 Návrh aplikace iOS



Obrázek 3.1: Diagram případů užití

Při návrhu aplikace jsem vycházel ze dvou modelových situací, ve kterých bude uživatel aplikaci používat, a požadavkům těchto situací jsem následně přizpůsobil návrh ovládání.

První ze situací je, že se uživatel aplikace nachází ve společnosti, kde dochází ke konzumaci alkoholických nápojů – například v restauračním zařízení. V tomto případě je uživatelským zájmem co nejjednodušeji vložit zkonsumovaný nápoj, případně ověřit dopad konzumace konkrétního nápoje na jeho stav v následujících minutách/hodinách. Například pro případ, kdy uživatel potřebuje zjistit, za jak dlouho dojde k vystřízlivění, aby mohl opět usednout za volant. Doba provádění těchto úkonů by měla být co nejkratší, protože v případě, že se nacházíme ve společnosti, může být obsluha telefonu považována za nezdvořilé gesto. Z tohoto důvodu byl vymyšlen koncept oblíbených nápojů, mezi které si uživatel přidá pár položek, které často konzumuje, a ty mu budou prezentovány ihned po spuštění aplikace. Jestliže se nápoj mezi oblíbenými položkami nenachází, i přesto je základní požadavek rychlosti vkládání stále platný, a proto kladu vysokou prioritu jednoduchosti vkládání záznamů. Vkládání záznamů by nemělo mít více kroků, než je nezbytně nutné. Druhým potenciálním úkonem v této situaci může být zjištění doby, kdy bude z krve

eliminován veškerý alkohol. Řešení, které navrhuji, je v podobě výrazného ukazatele, který by ihned a zřetelně podal informaci o momentálním stavu uživatele.

Druhá modelová situace, ze které vycházím, je, že uživatel chce získat informace o své předchozí aktivitě – ať už si potřebuje vzpomenout, co konzumoval předchozí den, nebo by rád získal přehled o své dlouhodobé konzumaci alkoholu pro diagnostické účely. V obou případech předpokládám, že tyto požadavky nejsou urgentní, proto nemusí zaujímat výsadní pozici na úvodní obrazovce po spuštění aplikace. Nicméně to neznamená, že by postrádaly důležitost nebo neměly by být rychle splnitelné.

3.1.2 Návrh aplikace WatchOS

Při návrhu aplikace pro hodinky je třeba odvíjet se od omezení, která jsou dána rozměry zařízení, výkonem a zejména možnostmi WatchKit. Malá obrazovka je nevhodná pro čtení většího objemu textu, tudíž není na místě snažit se uživateli poskytnout veškerý servis iOS aplikace. WatchOS aplikaci je třeba chápat jako rozšíření funkcionality iOS aplikace, nikoliv jako její plnohodnotnou náhradu. Jestliže se znovu zamyslíme nad modelovými situacemi popsány v podkapitole 3.1.1, pak nahrazení mobilní aplikace za aplikaci na hodinkách dává smysl pouze pro první z nich. Tudíž pro zopakování: „zájmem uživatele je co nejjednodušeji vložit zkonsumovaný nápoj, případně ověřit dopad konzumace konkrétního nápoje na jeho stav v následujících minutách/hodinách“. Není nic rychlejšího, než použít k tomuto účelu zařízení, které nosíte pořád na ruce. Hodinky by ze své podstaty měly především zobrazovat informace, které se váží k aktuální chvíli. Ať už zobrazují kolik je hodin, jaká je aktuální měsíční fáze, či teplota ovzduší. Proto, v kontextu mé aplikace, navrhuji zobrazit pouze aktuální koncentraci alkoholu v krvi a dobu potřebnou k vystřízlivění. Pro rychlé vkládání nápojů do databáze navrhuji opětovně použít koncept oblíbených nápojů, jelikož jeho užívání by mělo být uživateli známo již z aplikace iOS. V případě, že se zkonsumovaný nápoj nenachází mezi oblíbenými, měl by uživatel mít přesto možnost plně obsloužit vkládání do databáze za pomoci chytrých hodinek. Proto navrhuji způsob vkládání z kompletního seznamu nápojů, které aplikace podporuje.

3.1.3 Návrh barevné palety

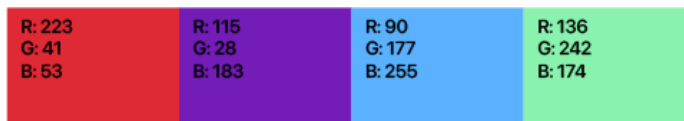
Volba barev uživatelského rozhraní může hrát větší roli, než jsme si často ochotni připustit. Špatná volba barevné palety může negativně ovlivnit rychlost vykonávání jednoduchých úkonů v aplikaci [12].

Vzhledem k předpokladu, že bude aplikace používána především ve večerních hodinách, zvolil jsem černou barvu pozadí. V noci nebude displej telefonu zbytečně oslňovat uživatele, a přitom nechá vyniknout světlejší prvky uživatelského rozhraní, které vykreslují sdělení s větší informační hodnotou. Barva pozadí tvoří dominantní barvu na každé obrazovce, proto dalším benefitem je, že na obrazovkách typu OLED¹, kde pro zobrazení černé barvy není třeba žádné energie, dochází k úspoře baterie.

Jako základní barvy aplikace jsem zvolil zelenou, modrou, fialovou a červenou. Touto volbou jsem dodržel definované základní principy, a to že uživatelské rozhraní by se mělo skládat maximálně ze čtyř barev. Větší množství barev by mohlo působit chaoticky a odvádět pozornost od podstaty sdělení. V případě této aplikace jsou barvy užívány jako grafické zkratky pro určení míry. Konkrétně každá barva označuje, jak velké procento alkoholu obsahuje nápoj nebo jak rizikově uživatel aplikace konzumoval alkohol v daný den. Pro

¹OLED – Organic light-emitting diode

srozumitelné vyjádření rozsahů pomocí barev by se zvolené barvy měly nacházet vedle sebe na barevném spektru [16], což je splněno. Zároveň užívám běžných asociací, kdy zelená barva je vnímána jako symbol něčeho bezpečného a červená barva běžně označuje něco nebezpečného, s vysokou intenzitou, případně konfliktního.

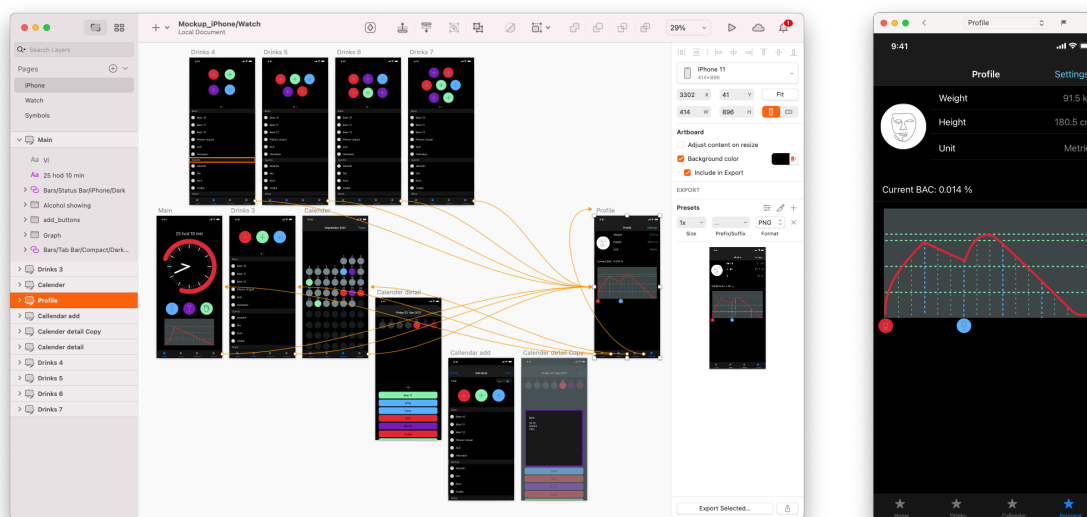


Obrázek 3.2: Barevná paleta použitá v aplikaci AlkoKolik

3.1.4 Prototypování

Prototypování by mělo být nedílnou součástí procesu vývoje mobilních aplikací. Vývojáři dává možnost rozvrhnout umístění prvků uživatelského rozhraní, a následně otestovat intuitivnost ovládání bez nutnosti implementace celé vnitřní logiky aplikace. V krátkém čase tak lze vytvořit návrh, který bude velmi podobný finálnímu produktu.

První verzi prototypu jsem vytvořil obyčejným nákresem na papír. V další fázi jsem se rozhodl pro použití aplikace Sketch², jelikož přeci jen umožňuje, narozdíl od tužky a papíru, využití dalších funkcionalit. Jedná se o vektorový grafický editor, který obsahuje pokročilé funkce pro snadné vytvoření kvalitního prototypu, jako je například propojení více obrazovek prototypu pomocí animovaného přechodu mezi obrazovkami po stlačení tlačítka. Důvodem volby tohoto softwaru byl fakt, že společnost Apple pravidelně publikuje grafické knihovny s vizuálními a ovládacími prvky běžně používanými v aplikacích ekosystému Apple jako plugin do této aplikace.

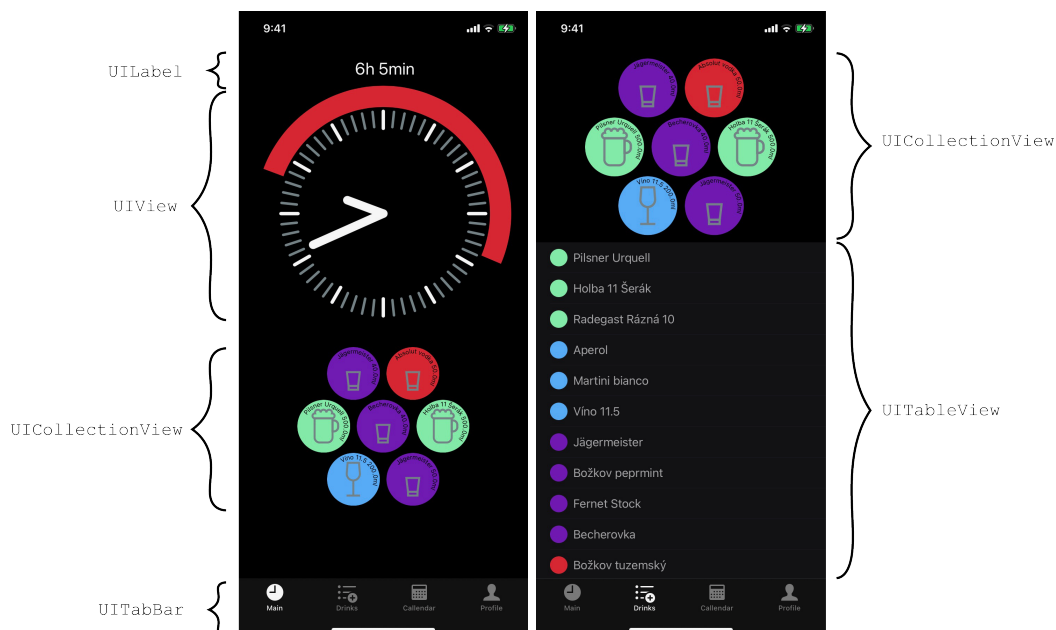


Obrázek 3.3: 1. prototyp aplikace AlkoKolik vytvořený ve Sketch

²aplikace Sketch B.V. Sketch – <https://www.sketch.com>

3.2 Uživatelské rozhraní iOS

Jako hlavní navigační prvek pro pohyb mezi obrazovkami aplikace jsem zvolil `UITabBar`. V iOS aplikacích je to běžně používaný, a tudíž pro uživatele snadno pochopitelný, způsob pohybu v aplikaci. V `UITabBaru` jsou umístěny čtyři záložky, jakožto reprezentace čtyř základních funkcionalit. Každé z nich je kromě názvu záložky přidělena ikona pro rychlejší orientaci.



Obrázek 3.4: Hlavní obrazovka a obrazovka vkládání nápojů

3.2.1 Hlavní obrazovka

Tato obrazovka je vstupní branou do aplikace. Jejím cílem je přehledně předat uživateli základní a nejdůležitější informaci – dobu, po kterou bude pod vlivem alkoholu. Vzhledem k tomu, že je to část aplikace, kterou uživatel uvidí nejčastěji, byl při jejím návrhu kladen velký důraz na srozumitelnost a jednoduchost. Uživatel není zahlcen přebytečnými informacemi, které by nebyly na první pohled zřejmé.

Dominantou této obrazovky jsou analogové hodiny s aktuálním časem, lemované kruhovým ukazatelem doby, po kterou bude uživatel aplikace pod vlivem alkoholu. Celý prvek je vytvořen pomocí *Core Animation*³, díky kterému mohou efektně animovat pohyby hodinových ručiček a změny doby střízlivění. Popis implementace se nachází v podkapitole 3.3.1. Fakt, že změny jsou vyobrazeny animovaně je pro uživatele získávání informací více komfortní. Doba, po kterou bude uživatel aplikace pod vlivem alkoholu, je nejdůležitější informace, která by měla být touto obrazovkou předána – proto je kruhový ukazatel červené barvy, která na rozdíl od ostatních nekontrastních vizuálních prvků působí výstražně. Pro úplnost a jednoznačnost je nad ciferníkem taktéž zobrazen `UILabel` s přepisem doby potřebné k dosažení nulové hladiny alkoholu v krvi.

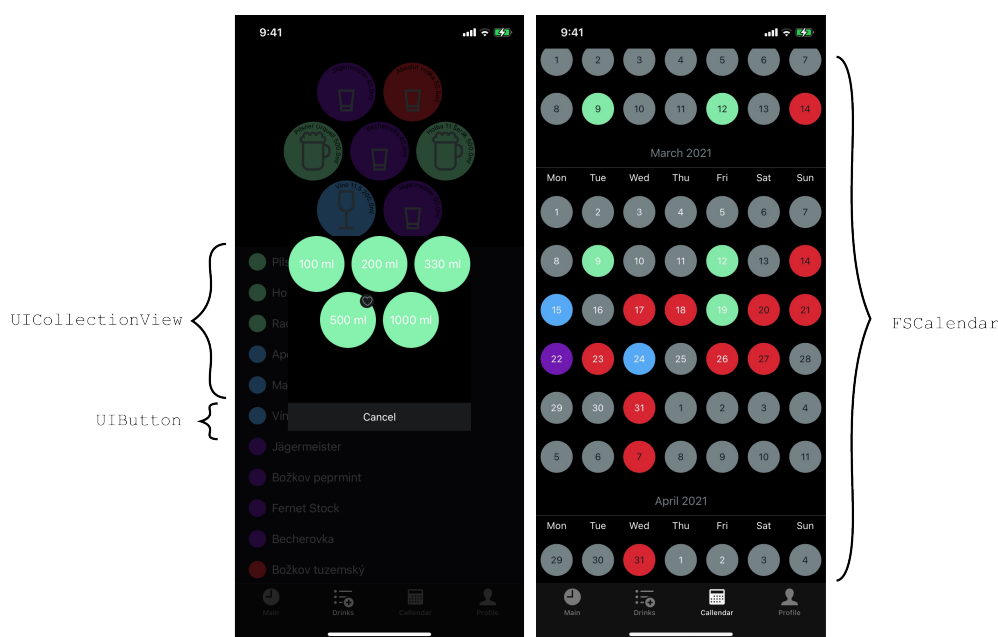
³framework Core Animation – <https://developer.apple.com/documentation/quartzcore>

Druhým prvkem, který je však volitelný, je `UIContainerView`, obsahující panel s tlačítky oblíbených nápojů. Cílem je uživateli umožnit rychlou volbu pro vložení zkonsumovaného nápoje do databáze. V krajním případě může mít uživatel v tomto panelu sedm tlačítek. Větší množství tlačítek by mohlo působit chaotickým dojmem a rušit tak uživatelský zážitek. Pro lepší orientaci je po obvodu tlačítka textový popis, na střed je umístěn piktogram nápoje, který má reprezentovat. Implementace tlačítek proběhla pomocí `UICollectionView`, jelikož je to běžně používané řešení pro kompaktní zobrazení obsahu, který lze tvořit podle jednotné šablony. Důvodem, proč je `UICollectionView` zaobaleno v `UIContainerView` je jednoduchá znovoupoužitelnost. Kontejnerové zobrazení mi umožňuje opětovné použití stejného ovládacího prvku na více místech v aplikaci bez nutnosti opakovaně programovat totožnou řídicí logiku.

3.2.2 Obrazovka vkládání nápojů

Druhou obrazovkou v pořadí `UITabView` je obrazovka, která slouží pro účely vkládání zkonsumovaných nápojů do databáze. Seznam všech nápojů řazený podle obsahu alkoholu je zobrazen v `UITableView`. Buňka v tabulce obsahuje název v podobě `UILabel` a kruhový indikátor, který je instancí `UIView` a je zbarven podle kategorie, do které daný nápoj patří. Po zvolení buňky je prezentována obrazovka „VolumeAlert“.

Nad tabulkou s výčtem nápojů je umístěno `UIContainerView`, zobrazující oblíbené nápoje obdobně, jako tomu je na hlavní obrazovce 3.2.1.



Obrázek 3.5: Obrazovka volby objemu nápoje a obrazovka Kalendář

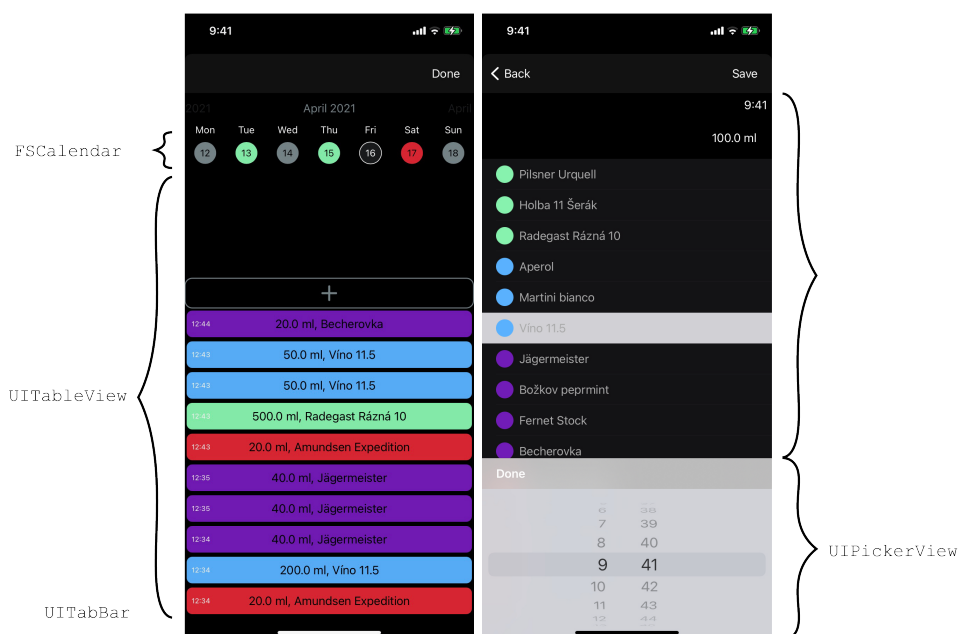
3.2.3 Obrazovka volby objemu nápoje

Než je vložen do databáze záznam o zkonsumovaném nápoji, tak je potřeba vědět jeho přesné množství. Za tímto účelem je uživateli prezentována tato obrazovka s oknem 3.6 obsahující `UICollectionView`. Tato kolekce používá stejný vizuální jazyk, jako kolekce s

oblíbenými nápoji popsaná v podkapitole 3.2.1 Každý nápoj se běžně konzumuje v jiných objemech, a proto každá buňka reprezentuje jeden objem.

3.2.4 Obrazovka Kalendář

Jak bylo zmíněno v podkapitole s názvem „Návrh aplikace iOS“ 3.1.1, vlastností aplikace by mělo být poskytnutí přehledu o zkonzumovaných alkoholických nápojích v delším časovém horizontu. Tato obrazovka obsahuje pouze měsíční kalendář, kde jsou jednotlivé dny opatřeny barevným indikátorem s informací o tom, jak moc se uživatel choval rizikově. Po zvolení dne v měsíci dojde k zobrazení modálního okna s „DayVC“. Kalendář je implementován pomocí doplňku FSCalendar⁴.



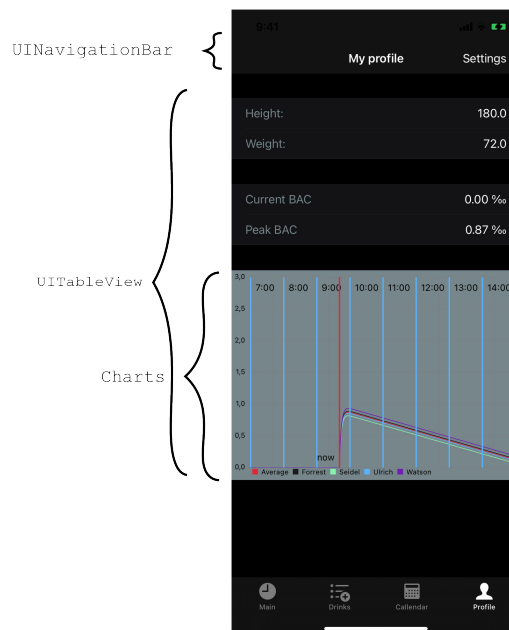
Obrázek 3.6: Obrazovka s náhledem dne a obrazovka pro přidávání nápojů v konkrétní čas

3.2.5 Obrazovka Náhled dne

Tato obrazovka slouží k poskytnutí podrobnějšího přehledu o konzumaci alkoholických nápojů ve zvoleném dni. K zobrazení přehledu používám UITableView, kde každý řádek UITableViewCell obsahuje název a množství zkonzumovaného nápoje společně s informací o čase vložení do databáze. Buňky jsou řazeny chronologicky, podle času vložení. V UITableView jsou povoleny úpravy, kdy uživatel gestem může odstranit řádek tabulky a změna je propagována do databáze. Zvláštností oproti běžné tabulce je, že buňky jsou zobrazeny od spodu.

V horní část se nachází FSCalendar se zobrazením dnů v týdnu a barevným indikátorem rizikovitosti chování, jak je popsáno v podkapitole Kalendář(3.2.4).

⁴knihovna FSCalendar – <https://github.com/WenchaoD/FSCalendar>



Obrázek 3.7:

3.2.6 Obrazovka osobního profilu

Poslední záložka v hlavním `UITabView` je věnovaná podrobným informacím o uživateli. Základ tvoří `UITableView` se třemi sekcemi a pevně daným počtem buněk.

V první sekci se zobrazuje informace o výšce a váze uživatele, které jsou získané za pomoci `HealthKit`. Jelikož jsou tyto informace použity při výpočtu predikce obsahu alkoholu v krvi, pokládám za důležité uživateli zobrazit z jakých předpokladů model vychází.

V druhé sekci `UITableView` jsou uživateli zobrazeny informace o aktuálním obsahu alkoholu v krvi na základě vypočteného modelu a předpokládané maximum koncentrace, kterého bude dosaženo.

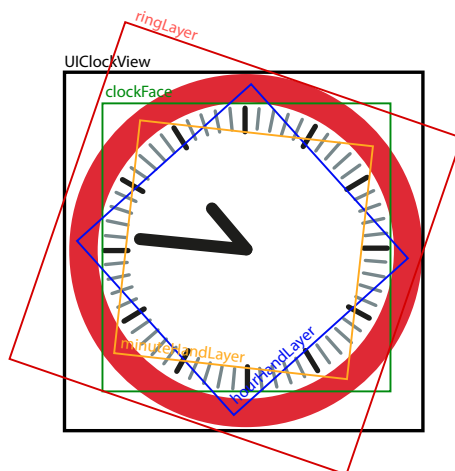
Samostatnou sekci tvoří graf, který zobrazuje průběh koncentrace alkoholu v krvi, jež jsou získány za použití všech zmíněných metrik v kapitole 2.1.3. Průběh alkoholemické křivky je vykreslen v časovém horizontu -5 dní až +2 dny. K vykreslení grafu byl použit `LineChart` z knihovny `Charts`⁵.

3.3 Implementace ovládacích prvků

3.3.1 Analogové hodiny

Při načtení úvodní obrazovky dochází k vykreslení hodinového ciferníku. Jedná se o samostatný `UIView`, jež v sobě kombinuje několik vizuálních vrstev `CALayer`. Každá vrstva reprezentuje část hodin, se kterou je potřeba manipulovat zvlášť. Proto samostatnou vrstvou je hodinový ciferník, minutová ručička, hodinová ručička a ukazatel doby vystřízlivění. Výhodou použití `CALayer` je, že se jedná o třídu z `Core Animation`, což je grafický framework, který se stará o vykreslování grafických prvků s využitím grafické karty. Díky tomu můžeme vykreslování vrstev bohatě animovat, vrstvy mohou obsahovat složité tvary

⁵knihovna `Charts` – <https://github.com/danielgindi/Charts>



Obrázek 3.8: Ukázka rozvrstvení hodinového cifníku

s velkým počtem vrcholů, vrstvy můžeme libovolně škálovat obsah, a přitom neklást zvýšené nároky na CPU.

Zvláštní podtřídou `CALayer` je `CAShapeLayer`, která reprezentuje ty prvky uživatelského rozhraní, jejichž tvar lze popsat pomocí Béziových křivek, k tomu slouží atribut `path`. Protože všechny prvky hodinového cifníku jsou geometrické tvary, byly použity pouze vrstvy odvozené od `CAShapeLayer`.

```
private lazy var watchHourFaceLayer : CAShapeLayer = {
    let layer = ClockCAShapeLayer()
    layer.fillColor = UIColor.clear.cgColor
    layer.strokeColor = UIColor.appWhite.cgColor
    layer.lineCap = .round
    layer.lineWidth = 4
    return layer
}()
.
.
.
init(){
    super.init(frame: .zero)
    layer.addSublayer(watchHourFaceLayer)
    backgroundColor = UIColor.clear
}
```

Výpis 3.1: Ukázka vkládání vrstvy do `UIView`

Skládání jednotlivých částí do výsledného `UIView` probíhá v následujícím pořadí. Nejprve je vytvořeno `UIView`. V rámci jeho inicializace jsou do jeho hlavní vrstvy přidány podvrstvy, reprezentující prvky hodin. Tyto podvrstvy mají nastavené pouze atributy týkající se vykreslování křivek, jako je tloušťka stopy, barva výplně apod., jak je možno vidět v ukázce 3.1. Až ve chvíli, kdy má dojít k vykreslení, typicky v `UIView` tuto chvíli reprezentuje metoda `layoutSubviews()` 3.2, dochází k definování hodnoty atributu `frame` podvrstev a

přiřazení cesty v podobě Bézierové křivky, protože až v tuto chvíli je známá velikost okna, do kterého se bude vrstva vykreslovat.

```
override func layoutSubviews() {
    super.layoutSubviews()

    watchHourFaceLayer.frame = bounds
    watchHourFaceLayer.path = createHourMarksPath(inside:
                                                    watchHourFaceLayer.bounds)
}
```

Výpis 3.2: Ukázka těla metody `layoutSubviews()`, kde dochází k ustanovení velikosti okna pro podvrstvu a vytvoření obrysu křivky, kterou bude zobrazovat

Po vykreslení `UIView` s hodinami se můžou naplno projevit všechny výhody implementace pomocí *Core Animation*. Od této chvíle lze plynule provádět animované změny obsahu ve vrstvách, jak je možno vidět v 3.8. Tento výstřížek kódů popisuje změnu ukazatele doby, kdy se mění velikost vykreslované výšeče z aktuální hodnoty na hodnotu $+1/60$. Jiný typ animovaného přechodu, který je použit, je animace rotace hodinových ručiček, která dodává pocit plynulosti uživatelského rozhraní.

```
func animateRingMinuteDown() {
    let animation = CABasicAnimation(keyPath: "strokeStart")
    animation.duration = 1.0
    animation.repeatCount = 1
    animation.byValue = 1/60
    backgroundRingLayer.strokeStart += 1/60
    backgroundRingLayer.add(animation, forKey: animation.keyPath)
}
```

Výpis 3.3: Ukázka těla metody `animateRingMinuteDown()`, kde dochází k animaci změny velikosti zobrazované výšeče kružnice

3.4 Uživatelské rozhraní WatchOS

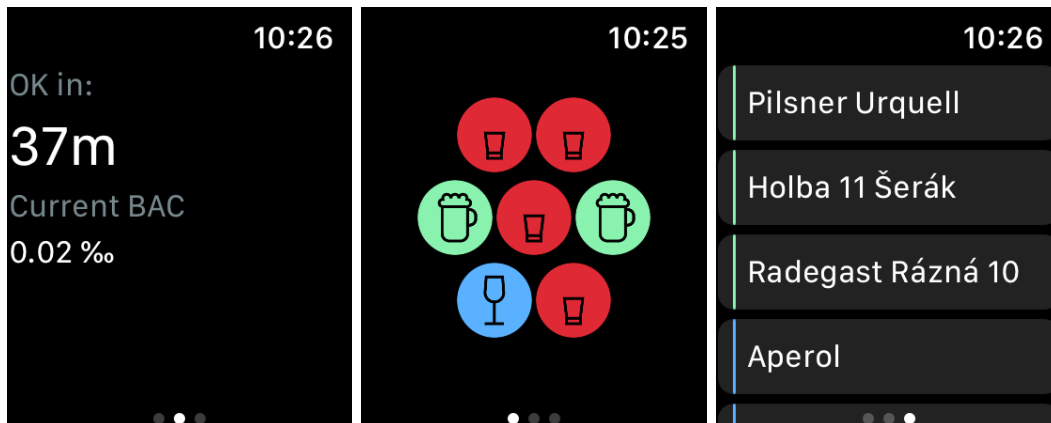
Z důvodů, které jsou zmíněny v podkapitole 3.1.2 se základ WatchOS aplikace skládá pouze ze tří obrazovek. Uživatel mezi nimi listuje pomocí gest, jedná se o takzvanou „page-based navigation“.

3.4.1 MainInterface

Tato obrazovka slouží jako úvod do aplikace. Pomocí `WKInterfaceLabel` zobrazuje informaci o aktuálním množství alkoholu v krvi a o času potřebném k opětovnému vystřízlivění. Každý z údajů je doplněn textovým popisem pro lepší orientaci uživatele.

3.4.2 FavouriteInterface

Cílem, který si klade tato obrazovka, je zobrazení oblíbených položek a umožnění uživateli s nimi interagovat. Jelikož knihovna `watchKit` neobsahuje třídu `UICollectionView` a ani žádnou jí podobnou třídu, která by dokázala efektivně řešit problém správného pozicování



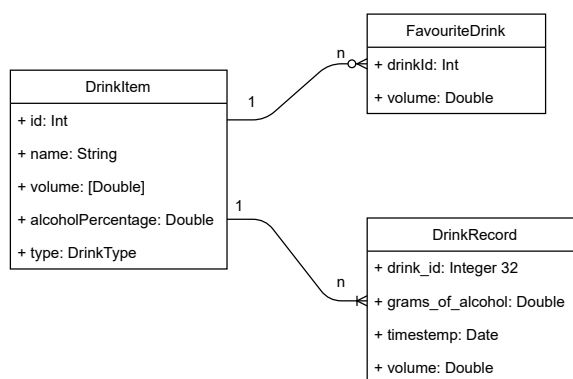
Obrázek 3.9: Obrazovky MainInterface, FavouriteInterface, AllDrinksInterface

proměnného počtu buněk současně ve vertikálním i horizontálním směru, musel být problém řešen staticky. V pohledu je umístěno sedm různých skupin tlačítek `WKInterfaceButton`, kde každá skupina reprezentuje různý počet nápojů, které jsou v oblíbených položkách. Aplikace následně uživateli zobrazí pouze tu skupinu tlačítek, která odráží počet položek, které jsou aktuálně oblíbené. V rámci skupiny jsem schopen správně umístit tlačítka, tak aby kopírovala vizuální jazyk iOS aplikace a uživateli nečinilo potíže se zorientovat.

3.4.3 AllDrinksInterface

Třetí v pořadí je obrazovka, která obsahuje tabulku `WKInterfaceTable` se seznamem všech dostupných nápojů. Každý řádek tabulky obsahuje `WKInterfaceLabel` zobrazující název nápoje a barevně odlišený `WKInterfaceSeparator` označující do které kategorie nápoj patří. Při zvolení řádku je uživateli prezentován `VolumeTableInterfaceController`, kde uživatel volí objem zkonsumovaného nápoje.

3.5 Datový model



Obrázek 3.10: Diagram databáze

3.5.1 Seznam nápojů

Předpokladem této tabulky je, že data v ní uložená budou uživatelsky neměnná. Z návrhářského hlediska neshledávám důvod, proč by měl uživatel mít potřebu upravovat, či mazat nápoje z nabídky aplikace. Zároveň svou povahou musí být dostupné ihned po startu aplikace, proto jsem navrhl seznam nápojů jako statickou tabulku, která je uložená v souboru a je nedílnou součástí aplikace. Pro její kódování byl využit formát JSON⁶, jelikož základní knihovna *Foundation*⁷ nativně podporuje kódování z formátu JSON. Řádky tabulky reprezentuje prvek JSON pole ve formátu 3.4. Pro případ, že by v budoucnu bylo nadbytečné zahrnovat nějaký nápoj do nabídky, byl přidán atribut **active**, který boolovskou hodnotou označuje, zda je nápoj aktivní, či nikoliv.

V rámci aplikace používám třídu **ListOfDrinksManager**, která zprostředkovává komunikaci mezi tabulkou a ostatními třídami, které vyžadují celý, nebo část jejího obsahu. Načtené položky z tabulky dále v aplikaci vystupují jako instance datové struktury **DrinkItem**.

```
{
    "id": 0,
    "name": "Pilsner Urquell",
    "volume": [100, 200, 330, 500, 1000],
    "percentage": 4.4,
    "type": "beer",
    "active": true
}
```

Výpis 3.4: Ukázka struktury JSON souboru uchovávající nápoje

U každého nápoje je uchována informace, do jaké kategorie alkoholických nápojů patří. Jak v instancích **DrinkItem**, tak i v JSON tabulce se pro uchování této informace používá atribut **type**. Kategorie, se kterými umí aplikace pracovat, jsou popisovány výčtovým datovým typem **DrinkType**, který může nabývat hodnot:

- **beer** – používáno pro nápoj vyrobený z ječného sladu, chmele a vody, kvašením za pomoci pivovarských kvasinek [8]
- **wine** – používáno pro kvašené ovocné nápoje s obsahem alkoholu minimálně 8,5%
- **cider** – myšleno jako kvašený ovocný nápoj z jablečného moštu s obsahem alkoholu 1,2%–8,5%
- **liqueur** – používáno pro alkoholické nápoje vyrobené z lihu nebo destilátu obsahující nejméně 100g cukru nebo ekvivalentní množství jiných sladidel na jeden litr výrobku[8]
- **spirit** – používáno pro alkoholické nápoje s vyšším obsahem alkoholu
- **cocktail** – používáno pro nápoje vzniklé smícháním několika různých většinou alkoholických nápojových surovin
- **none** – pro blíže neurčené nápoje

⁶JSON – JavaScript Object Notation

⁷framework Foundation – <https://developer.apple.com/documentation/foundation>

3.5.2 Záznamy o konzumaci

Záznam o konzumaci neboli `DrinkRecord` je jediná tabulka, která je implementována pomocí frameworku *Core Data*⁸. Při implementaci jsem vycházel z předpokladu, že aplikace bude uchovávat velké množství záznamů a nástroje, které poskytuje framework *Core Data*, jsou vhodné pro správu rozsáhlých databází.

Metody pro manipulaci s daty v databázi záznamů jsou součástí třídy `CoreDataManager`. Při vkládání záznamu do databáze metodou `CoreDataManager.insertRecord`, jsou hodnoty všech atributů `DrinkRecord` známy, až na hodnotu atributu `grams_of_alcohol`, který musí být dopočten. Pro přesný výpočet hmotnosti zkonzumovaného alkoholu používám vzorec (3.1)[10].

$$m = v * z * a * d \quad (3.1)$$

Kde:

m : je hmotnost požitého alkoholu v gramech

v : je objem zkonzumovaného nápoje v mililitrech

z : je podíl objemu ethanolu ku celkovému objemu, neboli procentuální zastoupení ethanolu v látce /100

a : je podíl absorbovaného alkoholu do lidského těla. Vždy se předpokládá, že se absorbuje 100% alkoholu, tudíž je při výpočtech použita hodnota $a = 1$

d : hustota ethanolu, která je konstantou o hodnotě 0,789 gramů na mililitr

3.5.3 Oblíbené nápoje

Vzhledem k nízkému maximálnímu počtu položek, které lze přidat v aplikaci mezi oblíbené, byla perzistence oblíbených nápojů implementována pomocí *UserDefaults*⁹. Je zde ukládána informace o jednoznačném identifikátoru nápoje a jeho objemu, které jsou zaobaleny do struktury `FavouriteDrink`.

3.6 Implementace simulačního modelu

Pro určení přesného času vystřízlivění je třeba provést simulaci průběhu alkoholemické křivky. Za tímto účelem byla vytvořena třída `SimulationAlcoholModel`, která seskupuje metody potřebné pro provedení simulace matematického modelu. Její instance je vytvořena při spuštění ve třídě `AppModel`, která se během životního cyklu aplikace stará o aktuálnost dat.

Výstupem simulace je pole hodnot datového typu `Concentration`, což je jinak pojmenovaný datový typ `Double`. Každý prvek pole na pozici n reprezentuje hodnotu koncentrace alkoholu v krvi v čase n (v minutách) od počátku simulace. Simulace se spouští metodou `SimulationAlcoholModel.run` a jako argumenty tato metoda přijímá základní údaje o uživateli (výška, váha, věk) zaobalené do struktury `PersonalData`, časové období, pro které má být sestaven simulační model, a volbu metody, která bude použita pro výpočet r faktoru 2.1.3. Samotný výpočet je rozdělen do tří fází. První fáze je pouze přípravná a v dalších dvou fázích dochází k výpočtu modelu. Důvodem rozdělení do fází je nutnost od sebe oddělit dva

⁸framework Core Data – <https://developer.apple.com/documentation/coredata>

⁹třída UserDefaults – <https://developer.apple.com/documentation/foundation/userdefaults>

procesy v lidském těle. První proces je vstřebávání alkoholu do krve a druhým procesem je odbourávání alkoholu z krve. Pro výpočet modelu tak vycházím z ideji, že koncentrace alkoholu v krvi v čase t je závislá na koncentraci alkoholu v krvi v čase $t-1$ a na přírůstku koncentrace v čase t .

V první fázi jsou získány všechny záznamy o konzumaci nápojů z databáze *Core Data* ve stanoveném časovém období a je vytvořeno pole pro uchování výsledků `resultArray`.

V druhé fázi výpočtu simulačního modelu se pouze zaobírám resorpcí alkoholu z nápojů do těla. Postupně je pro každý záznam `DrinkRecord` počítán průběh resorpční křivky metodou `calculateAbsorbtionPhases` 3.5. Takto získaný průběh je připočten k celkovému stavu koncentrace, který je uchován v `resultArray` a tak získáme celkový přírůstek koncentrace alkoholu v čase t .

```
func calculateAbsorbtionPhases(drink : DrinkRecord) -> [Concentration]{
    let dose : Double = drink.grams_of_alcohol
    let k_a : Double = 18
    var time : Double = 0.0

    var results : [Concentration] = []
    var c_old : Concentration = 0
    var c_new : Concentration = 0
    repeat {
        results.append(c_new-c_old)
        c_old = c_new
        time += 1
        let t = time / 60 // t je v hodinach
        c_new = dose * (1 - exp(-k_a * t))
    } while ((c_new/dose) < 1)
    return results
}
```

Výpis 3.5: Ukázka těla metody `calculateAbsorbtionPhases()`

V poslední fázi je počítána výsledná koncentrace alkoholu v krvi. Základ této fáze tvoří metoda `calculateEliminationPhases` 3.6, která na základě vstupní hodnoty aktuálního množství alkoholu v těle a aktuálního přírůstku množství alkoholu do těla vypočítá budoucí stav koncentrace látky v těle.

```
func calculateEliminationPhases(doseInBody: Concentration,
                               newAbsorbed: Concentration ,
                               person: AppModel.PersonalData,
                               method: RFactorMethod) -> Concentration{
    let r : Double = rFactorFor(method: method, person: person)
    let v_d : Double = r * person.weight.converted(to: .kilograms).value
    let timeChange : Double = 1/60

    let dose = doseInBody * v_d + newAbsorbed
    let c = (dose / v_d) - beta*timeChange
    return c > 0 ? c : 0
}
```

Výpis 3.6: Ukázka těla metody `calculateEliminationPhases()`

Kapitola 4

Testování

4.1 Testování uživatelského rozhraní

Testování uživatelského rozhraní probíhalo na skupině šesti uživatelů ve věkové kategorii 18–25 let. Předpokladem je, že majorita uživatelů aplikace AlkoKolik bude spadat do této kategorie. Nejdřív bylo uživatelům vysvětlena úvodní motivace a účel aplikace. Následně uživatelé obdrželi zařízení s čistou instalací aplikace a úkolem bylo seznámit se s uživatelským prostředím.

Pochopení pohybu mezi obrazovkami nečinilo problémy, ovšem dokud nebyl vložen první záznam o konzumaci, panovala všeobecná nejistota, kde hledat ukazatel času vystřízlivění. Co se týče úvodní obrazovky, tak respondenti pozitivně hodnotili animaci pohybu hodinových ručiček, nicméně při testování vyplynul nedostatek v podobě příliš dlouhého intervalu trvání animace, který byl následně upraven na pevnou hodnotu jedné sekundy. Při náhledu na graf s detailem průběhu alkoholemické křivky všichni respondenti kladli dotaz „co znamená legenda pod grafem“. Výpis a barevné odlišení metodik pro výpočet alkoholemické křivky je pro uživatele bez hlubší znalosti problematiky nesrozumitelné. Část uživatelů by uvítala bližší popis, o co se jedná, ale převážná většina uživatelů by uvítala odstranění této informace, protože je pro ně nadbytečná.

Z pozorování chování uživatele vyplynulo, že uživatelé při seznamování s uživatelským prostředím nevyzkoušeli všechny funkce. V následné diskusi uživatelé uváděli jako nejčastější důvod, že o funkcionalitě nevěděli. Proto v druhé fázi testování uživatelského rozhraní byly vysvětleny všechny funkce a princip jejich používání a opět došlo k pozorování chování. Uživatelé bez vysvětlení nebyli schopni přidat nápoj mezi oblíbené nápoje, ovšem po osvojení konceptu přidávání mezi oblíbené byla tato funkce hodnocena kladně. Dalším nedostatkem, který respondenti vnímali, bylo nedostatečné vysvětlení významu barevných indikátorů v tabulce s nápoji a v kalendáři. Intuitivně vnímali, že červeně zbarvená ikona značí „problém“, ale nebylo jim jasné, jaký problém tato barva zosobňuje. Oba výše zmíněné nedostatky související s nedostatečným pochopením funkcí aplikace byly vyřešeny vytvořením krátkého průvodce, který se uživateli zobrazí při prvním spuštění aplikace. Alternativní cestou pro zobrazení průvodce je vynucení jeho zobrazení v nastavení.

Celkově uživatelé hodnotili kladně přínos i použitelnost aplikace. Z testování uživatelského rozhraní vyplynou několik návrhů ke zlepšení, které by mohly být v budoucnu zapracovány.

- Lépe seskupit nápoje v seznamu nápojů. Jeden z respondentů by preferoval seskupení nápojů do sekcí podle kategorie.

- Přidat výpis podrobnějších informací o nápoji.
- Rozšířit seznam nápojů, které je možno zvolit.

4.2 Ověření platnosti na základě reálného měření

4.2.1 Pokus č. 1

Cíl Ověřit platnost modelu sestaveného aplikací AlkoKolik v reálných podmínkách.

4.2.2 Metodika

Skupina sledovaných osob se podrobí opakovanému měření koncentrace alkoholu v krvi při simulaci běžného společenského pití. Měření skutečných hodnot bude probíhat za pomoci alkoholtesteru značky V-Net Al 5500 s polovodičovým čidlem a deklarovanou přesností 10%.

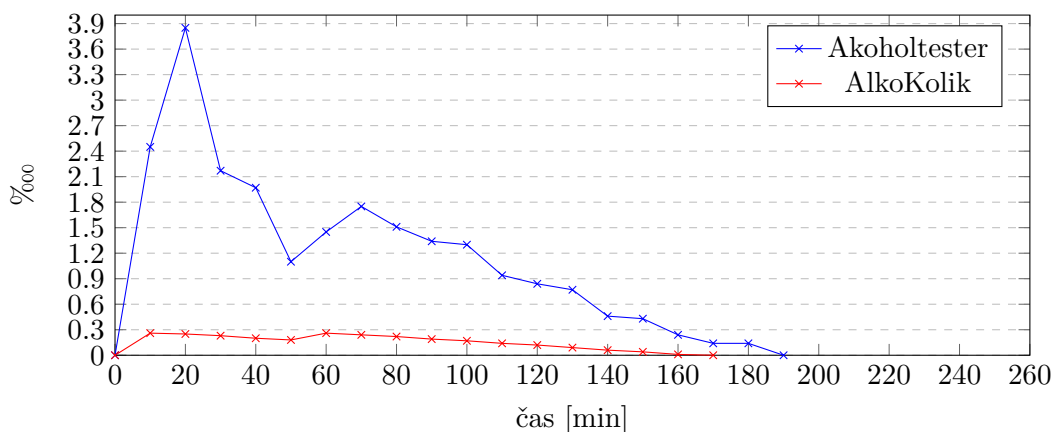
Postup

1. V čase 0 min proběhne měření koncentrace alkoholu v krvi.
2. Testované osoby vypijí alkoholický nápoj dle vlastního výběru obsahující minimálně 14 g alkoholu.
3. Každých 10 minut proběhne měření hladiny alkoholu v dechu alkoholtesterem.
4. Testované osoby v čase 55 minut od zahájení pokusu vypijí 20ml nápoje s obsahem 40% alkoholu.
5. Pokus je ukončen, jakmile množství alkoholu v dechu klesne na hodnotu 0‰.
6. Srovnáme naměřené výsledky s predikcí vytvořenou aplikací AlkoKolik.

Naměřené výsledky

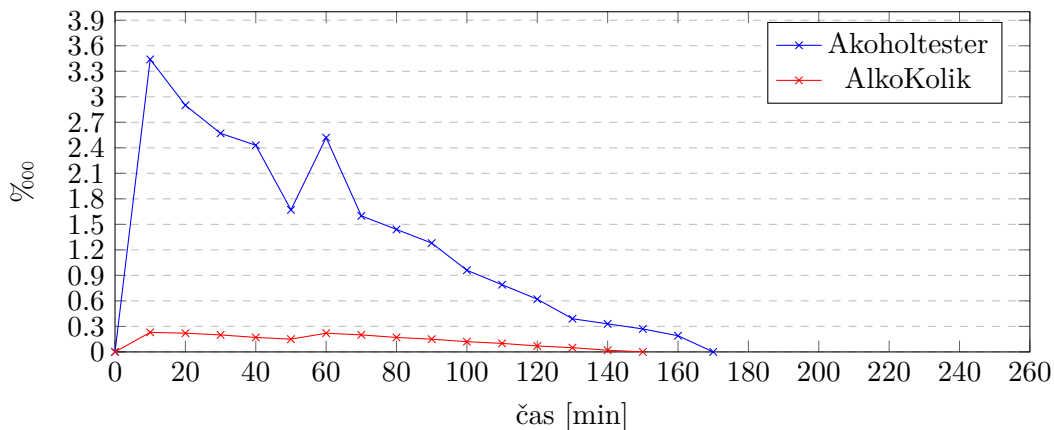
Osoba 1 Sledovaný subjekt č. 1 je muž ve věku 23 let, bez diagnostikovaných zdravotních problémů. Jeho výška v době uskutečnění testu je 180 cm a váha 76 kg. Prvním konzumovaným nápojem bylo pivo plzeňského typu s obsahem alkoholu 4,4% o objemu 500 ml.

Graf naměřených hodnot pro osobu č. 1



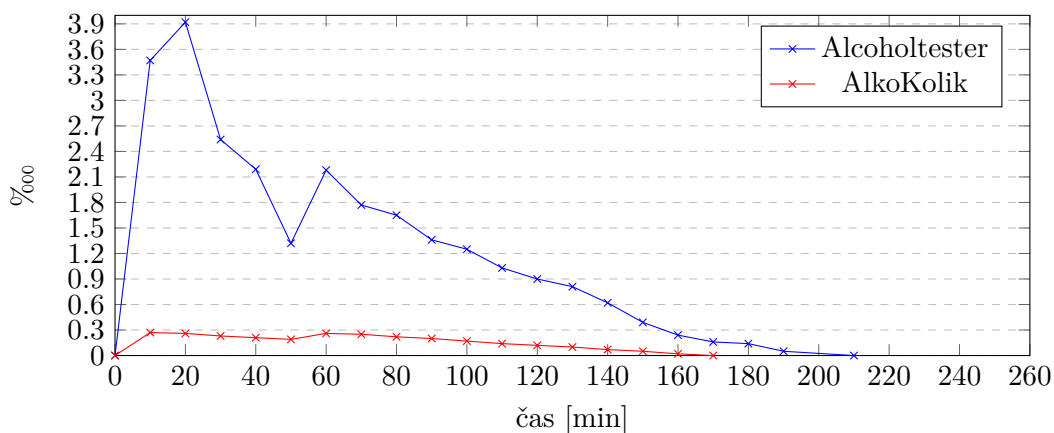
Osoba 2 Sledovaný subjekt č. 2 je muž ve věku 24 let, bez diagnostikovaných jaterních problémů. Jeho výška v době uskutečnění testu je 189 cm a váha 87 kg. Prvním konzumovaným nápojem bylo pivo plzeňského typu s obsahem alkoholu 4,4% o objemu 500 ml.

Graf naměřených hodnot pro osobu č. 2



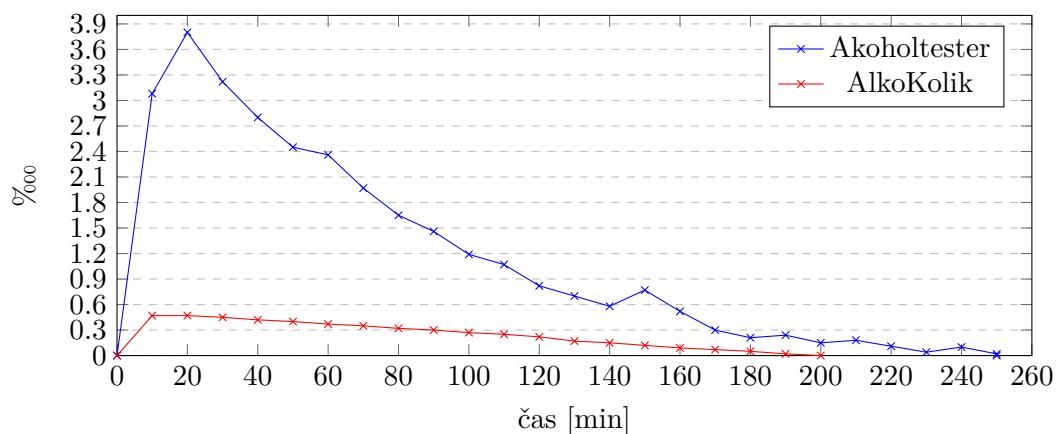
Osoba 3 Sledovaný subjekt č. 3 je muž ve věku 23 let, bez diagnostikovaných jaterních problémů. Jeho výška v době uskutečnění testu je 180 cm a váha 72 kg. Výběr prvního konzumovaného nápoje bylo pivo plzeňského typu s obsahem alkoholu 4,4% o objemu 500 ml.

Graf naměřených hodnot pro osobu č. 3



Osoba 4 Sledovaný subjekt č. 4 je ženského pohlaví ve věku 22 let, bez diagnostikovaných jaterních problémů. Její výška v době uskutečnění testu je 163 cm a váha 53 kg. Výběr prvního konzumovaného nápoje bylo víno s obsahem alkoholu 12% v objemu 200 ml. U této osoby bylo upuštěno od konzumace druhého nápoje v čase 55 minut od počátku měření.

Graf naměřených hodnot pro osobu č. 4



Tabulka 4.1: Souhrnná tabulka výsledků měření

| Čas | Osoba 1 | | Osoba 2 | | Osoba 3 | | Osoba 4 | |
|-----|----------|--------|----------|--------|----------|--------|----------|--------|
| | AkoKolik | Tester | AkoKolik | Tester | AkoKolik | Tester | AkoKolik | Tester |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 | 0.26 | 2.45 | 0.23 | 3.44 | 0.27 | 3.47 | 0.47 | 3.08 |
| 20 | 0.25 | 3.85 | 0.22 | 2.9 | 0.26 | 3.92 | 0.47 | 3.8 |
| 30 | 0.23 | 2.17 | 0.2 | 2.57 | 0.23 | 2.54 | 0.45 | 3.22 |
| 40 | 0.2 | 1.97 | 0.17 | 2.43 | 0.21 | 2.19 | 0.42 | 2.8 |
| 50 | 0.18 | 1.1 | 0.15 | 1.67 | 0.19 | 1.32 | 0.40 | 2.45 |
| 60 | 0.26 | 1.45 | 0.22 | 2.52 | 0.26 | 2.18 | 0.37 | 2.36 |
| 70 | 0.24 | 1.75 | 0.2 | 1.6 | 0.25 | 1.77 | 0.35 | 1.97 |
| 80 | 0.22 | 1.51 | 0.17 | 1.44 | 0.22 | 1.65 | 0.32 | 1.65 |
| 90 | 0.19 | 1.34 | 0.15 | 1.28 | 0.20 | 1.36 | 0.3 | 1.46 |
| 100 | 0.17 | 1.3 | 0.12 | 0.96 | 0.17 | 1.25 | 0.27 | 1.19 |
| 110 | 0.14 | 0.94 | 0.10 | 0.79 | 0.14 | 1.03 | 0.25 | 1.07 |
| 120 | 0.12 | 0.84 | 0.07 | 0.62 | 0.12 | 0.9 | 0.22 | 0.82 |
| 130 | 0.09 | 0.77 | 0.05 | 0.39 | 0.1 | 0.81 | 0.17 | 0.7 |
| 140 | 0.06 | 0.46 | 0.02 | 0.33 | 0.07 | 0.62 | 0.15 | 0.58 |
| 150 | 0.04 | 0.43 | 0 | 0.27 | 0.05 | 0.39 | 0.12 | 0.77 |
| 160 | 0.01 | 0.24 | 0 | 0.19 | 0.02 | 0.24 | 0.09 | 0.52 |
| 170 | 0 | 0.14 | 0 | 0 | 0 | 0.16 | 0.07 | 0.30 |
| 180 | 0 | 0.14 | | | 0 | 0.14 | 0.05 | 0.21 |
| 190 | 0 | 0 | | | 0 | 0.05 | 0.02 | 0.24 |
| 200 | | | | | 0 | 0 | 0 | 0.15 |
| 210 | | | | | | | 0 | 0.18 |
| 220 | | | | | | | 0 | 0.11 |
| 230 | | | | | | | 0 | 0.04 |
| 240 | | | | | | | 0 | 0.1 |
| 250 | | | | | | | 0 | 0.02 |
| 260 | | | | | | | 0 | 0 |

Závěr Počáteční odchylka mezi naměřenými hodnotami alkoholtesterem a výsledky z aplikace AlkoKolik bude způsobená tím, že se na stěnách dutiny ústní nacházely zbytky nevstřebaného nápoje. Výrobce podle návodu doporučuje, že by se měření mělo provádět minimálně třicet minut po konzumaci nápoje, aby se tomuto jevu zabránilo. Nicméně výsledky získané měřením se nedají považovat za důveryhodné. I po započtení tolerance deklarované výrobcem $\pm 10\%$ se výsledky reálného měření rozcházejí od očekávaného výstupu. Při měření osob alkoholtesterem se několikrát objevila hodnota vyšší než $0,5\text{‰}$, což není, podle dostupných zdrojů, reálné za žádných okolností.

Pokud bychom výstupy získané měřením alkoholtesterem interpretovali pouze jako binární hodnoty, kdy 0‰ znamená, že uživatel je střízlivý a ostatní hodnoty svedčí o opaku, pak můžeme hovořit o pozitivním přínosu této aplikace. Časový interval potřebný k vystřízlivění vypočítaný aplikací AlkoKolik a interval naměřený ve skutečnosti se od sebe neliší o víc jak 30% a v nejlepším případě se časy od sebe lišily pouze o 11% . Při přihlédnutí k množství neměřitelných faktorů, které aplikace AlkoKolik nemůže zahrnout do svého modelu, se pak jedná o uspokojivý výsledek měření.

Kapitola 5

Závěr

Cílem této bakalářské práce bylo vytvořit mobilní aplikaci a aplikaci pro chytré hodinky, do které bude uživatel zaznamenávat údaje o své konzumaci alkoholických nápojů. Aplikace bude na základě získaných dat vytvářet predikce průběhu koncentrace alkoholu v krvi, které budou prezentovány uživateli.

V první řadě bylo klíčové pochopit princip chování alkoholu v lidském těle. Bez znalosti faktorů ovlivňující vstřebávání a odbourávání by nebylo možné korektně předpovídat koncentraci alkoholu v těle. Dále došlo k návrhu prototypu a implementaci navrhnutého řešení, kde výsledkem jsou dvě funkční aplikace pro zařízení s iOS a WatchOS, které dávají uživateli informaci o aktuální hladině alkoholu v krvi, předpovídají průběh alkoholemické křivky a dávají možnost nahlédnout do uživateli historie konzumace alkoholických nápojů. Postupně vznikající verze aplikace byly podrobeny uživatelskému testování, s hlavním cílem zjištění nedostatku uživatelského prostředí. V rámci této etapy byly zjištěny drobné nedokonalosti v komfortu používání, které byly vždy zohledněny v dalším vývoji. Predikce koncentrace alkoholu v krvi z navrženého simulačního modelu, který je součástí aplikace, byly konfrontovány s reálným měřením alkoholtesterem. Vlivem velké nepřesnosti alkoholtesteru se nedá považovat provedené měření za objektivní, nicméně odpověď na základní otázku „zdali je člověk střízlivý či nikoliv“ poskytnout dokázalo. Při zkoumání pouze tohoto kritéria se časový údaj vystřízlivění podle aplikace a vystřízlivění na základě měření alkoholtesterem lišil maximálně o 30%.

Vzhledem k tomu, že téma bakalářské práce bylo mým nápadem, rád bych ve vývoji aplikace i nadále pokračoval. Je mi líto, že z důvodů mimořádného nařízení vlády nebyla možnost podrobit aplikaci testování v reálných podmínkách. Rád bych, po testování na větším vzorku uživatelů, aplikaci volně zveřejnil v App Storu pro širší veřejnost.

Literatura

- [1] APPLE INC.. *About the HealthKit Framework* [online]. Duben 2021 [cit. 2021-04-20]. Dostupné z: https://developer.apple.com/documentation/healthkit/about_the_healthkit_framework.
- [2] APPLE INC.. *IOS Design Themes* [online]. Duben 2021 [cit. 2021-04-20]. Dostupné z: <https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>.
- [3] COWAN, J. M., WEATHERMON, A., MCCUTCHEON, J. R. a OLIVER, R. D. Determination of volume of distribution for ethanol in male and female subjects. *J Anal Toxicol*. Sep 1996, sv. 20, č. 5, s. 287–290.
- [4] FORREST, A. The estimation of Widmark's factor. *J Forensic Sci Soc*. 1986, sv. 26, č. 4, s. 249–252.
- [5] GUSGÅRD, O. *Application development for the Apple Watch* [online]. 2018 [cit. 2021-05-09]. Bakalářská práce. Metropolia University of Applied Sciences. Vedoucí práce HJORT, P. Dostupné z: <http://urn.fi/URN:NBN:fi:amk-201805219250>.
- [6] HIRT, M. a NOVOMESKÝ, F. *Alkoholologie v soudním lékařství*. Martin SR: [b.n.], 2011. ISBN 978-80-8063-367-7.
- [7] JONES, A. W. Pharmacokinetics of Ethanol - Issues of Forensic Importance. *Forensic Sci Rev*. Jul 2011, sv. 23, č. 2, s. 91–136.
- [8] MINISTERSTVO VNITRA Česko. *Sbírka zákonů České republiky*. [online]. Praha : Tiskárna Ministerstva vnitra, 1993. ISSN 1211-1244. Dostupné z: <http://aplikace.mvcr.cz/sbirka-zakonu/>.
- [9] POSEY, D. a MOZAYANI, A. The estimation of blood alcohol concentration : Widmark revisited. *Forensic Sci Med Pathol*. Mar 2007, sv. 3, č. 1, s. 33–39.
- [10] SEARLE, J. Alcohol calculations and their uncertainty. *Med Sci Law*. Jan 2015, sv. 55, č. 1, s. 58–64.
- [11] SEIDL, S., JENSEN, U. a ALT, A. The calculation of blood ethanol concentrations in males and females. *Int J Legal Med*. 2000, sv. 114, 1-2, s. 71–77.
- [12] SHIEH, K.-K. a LIN, C.-C. Effects of screen type, ambient illumination, and color combination on VDT visual performance and subjective preference. *International Journal of Industrial Ergonomics*. 2000, sv. 26, č. 5, s. 527–536. DOI: [https://doi.org/10.1016/S0169-8141\(00\)00025-1](https://doi.org/10.1016/S0169-8141(00)00025-1). ISSN 0169-8141. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0169814100000251>.

- [13] ULRICH, L., CRAMER, Y. a ZINK, P. [Relevance of individual parameters in the calculation of blood alcohol levels in relation to the volume of intake]. *Blutalkohol*. May 1987, sv. 24, č. 3, s. 192–198.
- [14] WATSON, P. E., WATSON, I. D. a BATT, R. D. Prediction of blood alcohol concentrations in human subjects. Updating the Widmark Equation. *J Stud Alcohol*. Jul 1981, sv. 42, č. 7, s. 547–556.
- [15] WIDMARK, E. *Die theoretischen Grundlagen und die praktische Verwendbarkeit der gerichtlich-medizinischen Alkoholbestimmung*. Urban & Schwarzenberg, 1932.
[Fortschritte der naturwissenschaftlichen forschung ... n. f. Dostupné z: <https://books.google.cz/books?id=-TUbwEACAAJ>.
- [16] ŠPAČKOVÁ, Z. *Vizualizace uživatelského rozhraní [online]*. 2018 [cit. 2021-05-09]. Diplomová práce. Masarykova univerzita, Pedagogická fakulta, Brno. Vedoucí práce NOGA, P. Dostupné z: <https://is.muni.cz/th/f57bk/>.

Příloha A

Obsah přiloženého CD

Přiložené paměťové DVD má následující adresářovou strukturu:

```
./
├── AlkoKolik/ ..... složka se zdrojovými soubory aplikace
│                       iOS a WatchOS
├── AlkoKolik-BT-text/ ..... zdrojové soubory textu práce v TeXTeXu
├── AlkoKolik-thesis.pdf ..... text technické zprávy ve formátu PDF
├── README/
│   ├── README.md ..... text technické zprávy ve formátu
│   │                       Markdown
│   └── readme-img ..... ilustrační obrázky použité v README.md
```